Generating 'Distributed' Referring Expressions: an Initial Report

Generating 'Distributed' Referring Expressions: an Initial Report

Barbara Di Eugenio*
Computational Linguistics
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
dieugeni@andrew.cmu.edu

Johanna D. Moore
Department of Computer Science, and
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA, 15260, USA
jmoore@cs.pitt.edu

Carolyn Penstein Rosé
Computational Linguistics
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
Carolyn_Penstein_Rose@cs.cmu.edu

1 Introduction

Consider some possible ways in which S (the Speaker) may ask H (the Hearer) to bring her an ironing board that is located in the basement:

(1a) *Bring me the ironing board in the basement.*

(1b) *Bring me the ironing board from the basement.*

(1c) *Go down to the basement and bring me the ironing board.*

Arguably, only in (1a) is the location $\mathcal{L}$ of the ironing board (*the basement*) included in the NP whose head is *ironing board*.[1] In (1b), $\mathcal{L}$ is a modifier to the verb, *from the basement*. In (1c), $\mathcal{L}$ is just an *expectation*, derived from knowledge about performing a *bring* action, and from the relation between the two actions — *go down to the basement* is a substep in the plan that achieves *bring S the ironing board*.

In these examples, the referring expression in the NP is not sufficient to uniquely identify the intended referent, but it is its linguistic context that adds other necessary constraints. This is the reason why we call these referring expressions *distributed*. To our knowledge, while many researchers have worked on generating referring expressions, e.g. [Appelt, 1985], [Kronfeld, 1990], [Dale, 1992], [Pattabhiraman and Cercone, 1990],[2] distributed referring expressions have not been addressed yet.

Note that the whole linguistic context must be taken into account while generating (1c): this is shown by the redundant and infelicitous

(2) *Go down <u>to the basement</u> and bring me the ironing board <u>in the basement</u>.*

Besides *planning* knowledge about the action that a certain verb denotes, *lexical* knowledge about that verb also comes into play. Consider

* Di Eugenio and Penstein Rosé were partially supported by Carnegie Mellon Faculty Development Fund # 1-13709.

[1] In the appropriate context, (1a) could also be interpreted as asking H to bring the ironing board *into the basement*, where S is (or will be, at the time H executes the action). We currently neglect this possibility.

[2] Space limitations don't allow us to discuss other researchers' approaches.

the contrast between (1c), (3a) and (3b):

(3a) *Go down to the basement and take the ironing board with you*

(3b) *Go down to the basement and put the ironing board away*

Arguably the same abstract plan — that we'll refer to as *Fetch* — may achieve *bring x to y*, *take x to y* and *put x (away) into y*: namely, go to where x is, get hold of it, and take it to y. However, these verbs differ from a lexical semantics point of view, and different expectations are generated in the three cases. In (1c) H expects the ironing board to be in the basement: the expectation is strong enough that even if a second ironing board is either in sight or salient in the discourse when S utters (1c), H won't ground[3] the referent of the NP *the ironing board* to the one in sight (although he may presumably ask "Isn't that the one you need?"). In (3a), the NP *the ironing board* must refer in the current context, by being either visible or salient in the discourse. (3b) is yet a different case: H will either ground the NP *the ironing board* to the salient one, if there is one, otherwise he will delay such grounding until he is in the basement.

In this paper, we describe our first steps toward generating such referring expressions. The implementation and experimentation are currently underway.

2 Towards a solution

Our solution is based on the integration of a principled discourse planner, LONGBOW [Young and Moore, 1994; Young *et al.*, 1994], with a formalism that represents both lexical semantic knowledge about verbs and planning knowledge about the actions that the verbs denote [Di Eugenio, 1993; Di Eugenio and Webber, 1996].

LONGBOW is an extension to *partial-order causal link* (POCL) planners, in which a plan is represented as a set of partially-ordered steps connected by causal links. LONGBOW extends POCL planners by introducing action decomposition. The representation of a plan operator is divided into an *action operator*, that captures preconditions and effects, and a possibly empty set of *decomposition operators*: each decomposition operator represents a single layer expansion of a composite step, that provides a partial specification for the subplan that achieves the parent step's effects given its preconditions.

Representation Formalism. Di Eugenio's system is able to interpret examples such as (1c): it infers the relation between the actions in the two conjuncts[4] and the expectations under which the relation holds. Di Eugenio's system is crucially based on a formalism composed of two KBs. The first, the *action taxonomy*, stores lexical semantic knowledge about verbs — the semantic primitives are derived from Conceptual Structures [Jackendoff, 1990]; the second, the *plan library*, contains *recipes* [Pollack, 1986], i.e. common sense planning knowledge about the actions that the verbs in the action taxonomy refer to. Both are implemented in CLASSIC [Brachman *et al.*, 1991], a description logic based system: the terms defined in the action taxonomy are used in the recipes in the plan library. Among the defined recipes is *Fetch*.

Advantages of the two systems.
First, LONGBOW keeps clear track of dependencies among steps by means of *causal links*. If $step_1$ establishes an effect which is a precondition $p_{2,i}$ for $step_2$, a *causal link* is created between $step_1$ and $step_2$. If a third step $step_3$ has a precondition $p_{3,j}$ which is identical to $p_{2,i}$, and if no intervening step has undone $p_{2,i}$,[5] then another causal link will be established between $step_1$ and $step_3$, thus keeping clear track of dependencies among steps. This clarity can be exploited e.g., to avoid generating the redundant

[3]Note the difference between *referring* and *grounding*: an NP refers in the discourse, but it is grounded to an entity in the world.

[4]Di Eugenio's algorithm actually works on purpose clauses, rather than on purposive conjunctions.

[5]LONGBOW, being based on UCPOP, is able to resolve such *threats* when they arise.

(2), as discussed below.

Second, LONGBOW allows to distinguish between intended and side effects: an effect is intended if it plays a causal role in a plan [Young and Moore, 1994]. This is very important from a discourse planning point of view: first, the *intentions* of the speaker have been widely recognized as fundamental in both discourse interpretation and production [Grosz and Sidner, 1986; Moore and Pollack, 1993]. Second, if H reveals that an intended effect of previous discourse was not achieved, S should try to achieve that effect again; this is not the case for side effects.

Third, Di Eugenio's KR system allows expressing subtle lexical distinctions as the ones underlying *bring* and *take*. It also maintains a well specified connection between lexical definitions and the pertinent planning knowledge. Finally, there is a natural mapping of Di Eugenio's recipe representation to LONGBOW's, which facilitates their integration.

3 The generation process

Examples such as (1c), (3a) and (3b) can be generated by assuming that the first conjunct is a step in a plan to achieve the second conjunct.[6] Here, we focus on (1c) and on *intensional* descriptions of parameters: the expectation in (1c) arises because in the *Fetch* recipe, the initial location of the object to be moved, *the basement* in our examples, is defined only intensionally.

The parameter *depend* in INFORM-REF in Fig. 1 — one of the discourse operators we are experimenting with — is used when a parameter is described intensionally. If no such *depend*ency exists, a simpler INFORM-REF operator is invoked. In the *Action* operator of INFORM-REF, the actual description for *param* is computed by the function *unique-desc*. The description will be a conjunction of descriptors d_i; if a d_i consists of a unary predicate applied to *param* (SIMPLE(d_i)), no recursive call to INFORM-REF is necessary; however, if d_i is COMPLEX, i.e. describes a relation REL between *param* and other objects, INFORM-REF must be recursively called to provide a description for OTHER-PARAM(d_i).[7]

INFORM-REF establishes two effects: that the agent is able to identify *param*, and also able to identify *depend*. Thus, when the planner generates the first conjunct α = *Go down to the basement*, in the context of the plan for β = *bring S the ironing board*, INFORM-REF$_\alpha$,[8] is invoked to generate the description *the basement*: it establishes that the agent is able to identify the basement, and also that the agent recognizes the dependency between the basement and the object to be moved. When planning to generate β, INFORM-REF$_{\beta_1}$ is called to generate the description *the ironing board in the basement*, and thus, INFORM-REF$_{\beta_2}$ is recursively called to establish the effects that the agent identifies *the basement*, and the locational dependency between *basement* and *ironing board*. However, such effects have already been established, so this step won't be expanded; rather, the appropriate causal link will be generated,[9] thus "explaining" why the expectation that the ironing board is in the basement comes about. Thus, no redundant description of the kind illustrated in (2) will be generated. Note that out method doesn't rest on α being generated before β: if the order were reversed, INFORM-REF$_{\beta_2}$ would be expanded, and INFORM-REF$_\alpha$ wouldn't. However, the appropriate causal links would still be established.

4 Conclusions

We see our work as synthesizing several themes: the generation of referring expressions; the spe-

[6] We haven't addressed yet *why* the system decides to expand the domain plan in this way.

[7] In this case, we must avoid infinite regress, that may arise when using relations in deriving referring expressions, as discussed by [Dale and Haddock, 1991].

[8] The subscript is meant to distinguish different instantiations of INFORM-REF in the plan.

[9] The new causal link will link INFORM-REF$_\alpha$ and whatever other step would have been using INFORM-REF$_{\beta_2}$'s effects as preconditions.

Action	
Header:	INFORM-REF(S, H, param, depend)
Preconditions:	$\neg$ ABLE(H, identify(param))
Effects:	ABLE(H, identify(param)) $\wedge$ ABLE (H, identify (depend))

Decomposition	
Header:	INFORM-REF(S, H, param, depend)
Constraints:	$\exists$ unique-desc(H, param, depend, $d_1 \wedge ... \wedge d_n$)
Steps:	Start, $\forall$ d_i, SIMPLE(d_i): INFORM(S, H, d_i), $\forall$ d_i, COMPLEX(d_i): INFORM(S, H, REL(d_i)), INFORM-REF(S, H, OTHER-PARAM(d_i), d_i), Final

Figure 1: The INFORM-REF operator

cific demands of *instructional text*: and issues of Knowledge Representation, among which the need to represent both *lexical* and *planning* knowledge about action verbs.

We still have quite some work to do. The integration of the two systems is currently underway, and so is the implementation and refinement of the discourse operators (some of the domain operators, are those already defined in Di Eugenio's KR system). We also have to add *bring* and *take* to the action taxonomy.

References

[Appelt, 1985] Douglas Appelt. *Planning English Sentences.* Studies in Natural Language Processing. Cambridge University Press, 1985.

[Brachman *et al.*, 1991] Ronald Brachman, Deborah McGuinness, Peter Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. Living with CLASSIC: When and How to Use a KL-ONE-like Language. In John F. Sowa, editor, *Principles of Semantic Networks — Explorations in the Representation of Knowledge*, pages 401–456. Morgan Kaufmann, 1991.

[Dale and Haddock, 1991] Robert Dale and Nicholas Haddock. Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252—265, 1991.

[Dale, 1992] Robert Dale. *Generating Referring Expressions.* ACL-MIT Series in Natural Language Processing. The MIT Press, 1992.

[Di Eugenio and Webber, 1996] Barbara Di Eugenio and Bonnie Lynn Webber. Pragmatic Overloading in Natural Language Instructions. *International Journal of Expert Systems, Special Issue on Knowledge Representation and Reasoning for Natural Language Processing*, 9(2), March 1996. To appear.

[Di Eugenio, 1993] Barbara Di Eugenio. *Understanding Natural Language Instructions: a Computational Approach to Purpose Clauses.* PhD thesis, University of Pennsylvania, December 1993. Technical Report MS-CIS-93-91.

[Grosz and Sidner, 1986] Barbara Grosz and Candace Sidner. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12:175–204, 1986.

[Jackendoff, 1990] Ray Jackendoff. *Semantic Structures.* Current Studies in Linguistics Series. The MIT Press, 1990.

[Kronfeld, 1990] Amichai Kronfeld. *Reference and Computation.* Studies in Natural Language Processing. Cambridge University Press, 1990.

[Moore and Pollack, 1993] Johanna Moore and Martha Pollack. A Problem for RST: the Need for Multi-Level Discourse Analysis. *Computational Linguistics*, 18(4):537–544, 1993.

[Pattabhiraman and Cercone, 1990] T. Pattabhiraman and Nick Cercone. Selection: Salience, relevance and the coupling between domain-level tasks and text planning. In *Proceedings of the Fifth International Workshop on Natural Language Generation, Pittsburgh, Pennsylvania*, pages 79–86, 1990.

[Pollack, 1986] Martha Pollack. *Inferring Domain Plans in Question-Answering.* PhD thesis, University of Pennsylvania, 1986.

[Young and Moore, 1994] R. Michael Young and Johanna D. Moore. DPOCL: A Principled Approach to Discourse Planning. In *Seventh International Workshop on Natural Language Generation*, pages 13–20, Kennebunkport, Maine, 1994.

[Young *et al.*, 1994] R. Michael Young, Martha E. Pollack, and Johanna D. Moore. Decomposition and causality in partial order planning. In *Second International Conference on Artificial Intelligence and Planning Systems*, 1994.